Changes to the Specification

Please replace paragraph [0038] with the following amended paragraph:

[0038] To process a driver request of a loaded driver, the processor 120 determines whether the driver request is associated with a violation condition of a protocol interface to prevent the protocol interface from being called on inadvertently or maliciously. In the example of flow chart 500 in FIG. 5, the processor 120 may determine if the driver request is associated with a violation condition by determining whether the driver request is a request to access a restricted protocol interface (e.g., a platform password protocol previously installed to the processor system 100) by the loaded driver (block 510). In particular, the processor 120 may process a handle protocol associated with the loaded driver. As noted above, the handle protocol associated with the loaded driver points to a list of one or more protocols configured to respond to requests for services. The loaded driver may attempt to access the restricted protocol interface. For example, the restricted protocol interface may be accessible by only one caller such as the phase core 330 to prevent the restricted protocol interface from being corrupted. Accordingly, the processor 120 rejects the request (e.g., protects the restricted protocol interface from the loaded driver by storing the restricted protocol interface in a protocol database (i.e., hiding the restricted protocol interface from the loaded driver)) (block [[515]]525). For example, the protocol database may be a data structure stored in the main memory 130 and/or one or more mass storage devices 180. The protocol database may be configured to store one or more restricted protocol interfaces.

Please replace paragraph [0041] with the following amended paragraph:

[0041] While FIG. 5 is depicted to include blocks 510, 520, and 530 to determine if the loaded driver is associated with a violation condition, the processor 120 may also determine if the loaded driver is associated with a violation condition by implementing only one of blocks 510, 520, and 530. Referring to flow chart 600 of FIG. 6, for example, the processor 120 may determine whether the driver request is a request by the loaded driver to access a restricted protocol interface installed in the processor system 100 (block 610). If the driver request is a request by the loaded driver to access a restricted protocol interface, the processor 120 may reject the request (e.g., protect the restricted protocol interface from the loaded driver by storing the restricted protocol interface in a protocol database (i.e., hiding the restricted protocol interface from the loaded driver)) (block [[615]]625).

Please replace paragraph [0045] with the following amended paragraph:

[0045] In another example, the processor 120 may determine whether the driver request is a request to access a restricted protocol interface (block 1010) and determine whether the driver request is a reinstall request (block 1020) to determine if the loaded driver is associated with a violation condition as shown in flow chart 1000 of FIG. 10. The processor 120 may store the restricted AP in the protocol database ~~(block 1015)~~ or reject the reinstall request by the loaded driver (block 1025).

Please replace paragraph [0046] with the following amended paragraph:

[0046] As yet another example, the processor 120 may determine whether the driver request is a request to access a restricted protocol interface (block 1110) and determine whether the driver request is an install request by the loaded driver (block 1130) to determine if the loaded driver is associated with a violation condition as shown in flow chart 1100 of FIG. 11. The processor 120 may store the restricted protocol interface in the protocol database ~~(block 1115)~~ or reject the install request by the loaded driver (1125). As a result, the restricted protocol interfaces of the processor system 100 (i.e., the APs 320) are protected from unauthorized access (e.g., access by callers other than the phase core 330).